UNCLASSIFIED

AD NUMBER
AD012913
LIMITATION CHANGES
TO: **Approved for public release; distribution is unlimited.**
FROM: **Distribution authorized to DoD only; Administrative/Operational Use; 02 APR 1953. Other requests shall be referred to Naval Ordnance Laboratory, White Oak, MD.**
AUTHORITY
USNOL ltr dtd 29 Aug 1974

THIS PAGE IS UNCLASSIFIED

NAVORD REPORT 2755

MANUAL OF PROGRAMMING FOR THE CARD-PROGRAMMED CALCULATOR

2 APRIL 1953

U. S. NAVAL ORDNANCE LABORATORY
WHITE OAK, MARYLAND

NAVORD Report 2755
Aeroballistic Research Report No. 153

MANUAL OF PROGRAMMING
FOR THE CARD-PROGRAMMED CALCULATOR

Prepared by:

Benjamin F. Cheydleur, Calvin C. Elgot,
Elise R. Fisher, Carolyn H. Gruner,
Walter F. Malmborg, Hoyt L. Stevens,
Thomas S. Walton

ABSTRACT: A brief description of the Card-Programmed Calculator is given. Instructions on how to program for the machine as used with the NOL General Purpose Panel are given in detail. Thus a beginner is able to program by becoming familiar with this manual.

U. S. NAVAL ORDNANCE LABORATORY
WHITE OAK, MARYLAND

This report summarizes in useful form a development over the last two years which has culminated in a very flexible and high speed system of directing a card-program electronic computor. The system permits two instructions per cycle to be completed and a wide range of transcendental functions to be computed in a fraction of a second. The work was carried out under Project NOL-Re9a-108-1-53, entitled "Aerodynamics and Fluid Mechanics." The results have made possible the automatic solution of a wide variety of problems in connection with ordnance research and development.

EDWARD L. WOODYARD
Captain, USN
Commander

H. H. KURZWEG, Chief
Aeroballistic Research Department
By direction

CPC MODEL II MANUAL

TABLE OF CONTENTS

CPC MODEL II MANUAL

TABLE OF CONTENTS

Continued

MANUAL OF PROGRAMMING
FOR THE CARD-PROGRAMMED CALCULATOR

I INTRODUCTION

The presentation of a mathematical problem to a computing instrument consists of the steps:

1. Problem creation and recognition.
2. Detailed problem definition in complete and condensed mathematical form.
3. Application of the methods of numerical analysis to establish specific computational formulae which will yield numerical approximations to that mathematical form and will assure an economy of procedure commensurate with the accuracy required.
4. Translation of the resulting discrete statement of the problem into sequences of algebraic steps which are connected together by means of branch operations which depend on the computed outcome of preceding sequences.
5. Formulation of a logistical pattern of storage to assure availability of the basic input and of the intermediate results which occur at every step in the sequence of computation.
6. Formulation of a useful arrangement and pattern of presentation of the important results.
7. Translation of the individual algebraic steps and branches into code, which is the language to which the computing instrument responds.

Although all of these steps may be considered by a problem programmer, the outline herein is confined chiefly to steps 5, 6, and 7, with a few remarks on step 4. This is in consideration of the main intention, which is to provide a short manual of coding principles applicable to the Card-Programmed Calculator, Model II, as it is controlled by the so called General-Purpose Double-Operation Function Panel, developed in the Naval Ordnance Laboratory.

Accordingly much of the material is presented in the form of directions to the programmer, beginning at a stage in the problem development somewhere between steps 4 and 5 above. At this stage a systematic record of the algebraic and other steps in the sequence is produced and corresponding codes are assigned. This record is the <u>program sheet</u>, shown as attachment 1) From it, the actual instruction and data cards to be fed into the machine are prepared. An instruction card design is shown in attachment 2).

II DESCRIPTIONS

1. The Card

1.1 The basic operating unit of IBM machines is the IBM punched card, which may contain both coded instructions and data in the form of punched holes. The card has 80 vertical columns, each with 12 punching positions, one each for the digits 0 to 9, an 11 or X position, and a 12 or Y position. A blank column is interpreted as a zero. The card is divided into sections, or groups of columns, called fields; these define portions of the card in which information of a certain kind will always appear

1.2 For use with the General Purpose Panel the card is divided into 10 fields. Fields numbered 1-7 are for data and include respectively columns 3-10, 27-35, 36-44, 45-53, 54-62, 63-71, and 72-80. In fields 2 to 7 the first 8 columns are for digits and the ninth column is the sign, blank or zero punch for a positive sign, "9" punch for a negative sign. In field one, column 3 is the sign column and the seven columns 4 to 10 are for digits. These seven card fields may all be used at the same time.

1.3 Columns 13-26 are allocated for instruction coding.

1.4 Columns 11 and 12 are allocated for sequential card numbering.

1.5 Columns 1 and 2 are for the identification of the problem, called a project number or MARR number. Every card used in connection with a problem must have this project number punched in it. Columns 1 and 2 are not read by the machine.

2. The Program Sheet

The program sheet is a detailed layout of a calculation. This is a form consisting of pairs of wide and narrow horizontal spacings. Each such pair is called a line. In the wide spacings, the algebra corresponding to each computation step is written by the programmer. Immediately above this, within the corresponding narrow spacing, the programmer can then code the actual decimal digits corresponding to that algebra, to the line number, and to the other identification he chooses to be useful. Thus each line should contain the codes to be punched in one card for the algebraic and arithmetic manipulations which that card will cause the calculator to perform, and the numerical values to be read into the registers from that card.

3. The Components of the Card-Programmed Calculator

3.1 The four basic components of the Card-Programmed Calculator are: one type 412 Accounting Machine, one type 605 Electronic Calculator Unit, one type 527 Gang-Summary Punch Machine and one type 941 Auxiliary Storage Unit. (See attachment 3.)

The functions of the individual components are as follows:

3.2 The 412 Accounting Machine serves as card-reader, printer, electromechanical storage and control. Through its selector and comparing relay circuits, all coded instructions are translated into individual actions (e.g., operations for the 605 calculator, summary punch orders for the 527, orders for printing, spacing, shifting, etc.).

All input data is read into the Card-Programmed Calculator by way of the 412 machine. Four fields of this data may be read directly from the card into four storage units of the 412. Simultaneously, three operands of data may be read into three registers of the 605 electronic unit from the card. There is a total of seven storage units in the 412 machine but only four of these may receive information directly from the card. Each of the seven units can receive results from the 605 via channel "C" (See attachment 3), store them, and transmit them back into the 605 registers via channels "A" and "B" for later use. Six of these units have nine positions of storage, eight for digits and one for the sign. One unit has seven positions for digits and one for sign.

3.3 The 941 auxiliary storage unit consists of sixteen electromechanical registers which receive numbers from channel "C" and transmit numbers to channels "A" and "B." Each register has nine positions of storage, eight for digits and one for the sign.

3.4 The 605 electronic calculator unit uses electronic circuits to perform the operations called for by instruction cards.

3.4.1 With each card cycle of the 412, the 605 unit completes sixty consecutive electronic programs, each one of which can be used to perform a simple operation such as a division, multiplication, shift, transfer, etc. By the selection of groups of these programs, the performance of the 605 can be varied from card cycle to card cycle. This selection is made by the 412 which translates the punched instruction code into a selector (relay) configuration which is used to suppress or alter programs.

3.4.2 The arithmetic unit interprets a sequence of eight digits in any register or storage as a decimal number with the decimal point occurring after the high order digit, i.e., a sequence of eight digits XXXXXXXX is interpreted as X.XXXXXXX.

3.4.3 Normally after the sixtieth program is completed, the 412 reads another card and the process is repeated, but by energizing the "program repeat" device, the 605 will continue to recycle through the sixty programs until the device is suppressed. This device is used for producing transcendental functions which are computed in the form of power series and each term is developed by a sweep of sixty programs.

3.4.4 The electronic storage registers of the 605 unit have been arranged to form four eight position registers. Three of these are "read-in" registers for the operands, U, V, and W. The fourth is a read-out register for the intermediate result Y. The final result Z is read out of the arithmetic unit of the 605. Either one of these results may be sent via channel "C" to a storage unit.

3.4.5 For the read-in registers in the 605 two channels ("A" and "B") are provided to transmit numbers from the 412 and 941 storages of the Card-Programmed Calculator. Operand U is read via channel "A" and operand V or W is read via channel "B." Only one channel serves both V and W. Hence, it is not possible to call quantities from these storages into both V and W on the same card cycle.

3.4.6 Another facility of the 605 is the balance test unit which tests the sign of the result of a calculation. On the basis of the result of this test, the 605 can then suppress or permit subsequent operation for the control of sub-routines, that is, the test can cause the suppression of operations coded on a set of instruction cards.

3.5 The 527 punch machine is used to punch results on cards. These can be used as input data cards for new phases or iterations of problems. The contents of any of the 412 storages and of the 605 registers W, Y, and Z can be punched simultaneously in a new card. This action is controlled by a coded instruction read by the 412.

4. The Teleplotter

4.1 The Teleplotter is a digital plotting device that will convert a digital input of point coordinates to an (x,y) plotting head position by actually counting the lines on the graph paper. The essential components of the Teleplotter are a flat plotting area on which is placed the coordinate paper; a plotting head which travels in two orthogonal directions and covers the entire field of the plotting area; one decimal electronic accumulator for each axis; and the necessary control elements.

4.2 Although the Teleplotter is not an IBM machine, it has been built to work in conjunction with such machines. When it is connected with the CPC the operation of the Teleplotter is automatic and is under the control of the instruction cards passing thru the CPC. The proper instructions will cause a transfer of numbers from a CPC storage to the Teleplotter memory; plotting of the desired point will proceed automatically while the CPC is continuing with further calculations.

III ACTION AND INTERRELATION OF THE PARTS: A. INPUT

5. Addresses

5.1 Addresses are the codes by which operands are read into the registers U, V, and W and by which results are directed to the mechanical storages for later use.

5.1.1 The address code, or more simply the address, consists of two digits which are punched in columns 14-15, 17-18 and 20-21 when referring to the U, V, and W operands respectively; and in columns 22-23 when directing a result to mechanical storage.

5.1.1.1 A storage address in columns 22 and 23 normally indicates that Z is to be stored. However, an "X" punch in the units position of the storage address (column 23) causes Y to be stored and/or printed instead of Z. (See 9.1.5)

5.1.1.2 The final card in the 412 card feed will not store a result.

5.1.2 Addresses are of four general types: those referring to the 412 and 941 electromechanical storages, the electronic storages, and the fields 5, 6, and 7 of the card.

5.2 The 941 storage contains two banks of eight registers each. The address codes are 11, 12, ... 18 (for bank 1) and 21, 22, ... 28 (for bank 2). These two digit numbers may be the addresses for the operands U, V, and W via channels "A" and "B" and for storing the results Y or Z via channel "C."

5.2.1 These storage units always reset to zero just before a number is read into them.

5.2.2 A number must not be recalled from a 941 storage unit prior to the third card following the card on which it was addressed to that storage unit.

5.3 The 412 counter storage responds to addresses 71, 72, ... 77. These are to be used for sending Y or Z to storage via channel "C," and for entering the U, V, and W registers via channels "A" and "B" without resetting the corresponding storage units.

5.3.1 81, 82, ... 87 control the same respective units, but are used for U, V, and W, in place of 71, 72, ... 77 to cause resetting of the unit after it has read out.

5.3.2 A number must not be recalled from a 412 storage unit prior to the second card following the card on which it was addressed to that storage unit.

5.3.3 The counters in the 412 storage are wired to accumulate positive numbers only.

5.3.4 Results may be read into the same 412 storage unit on successive cards.

5.3.5 In order to read from a single 412 storage unit into two operand registers on the same card and reset, the tens digit of both addresses must be 8.

5.3.6 An "X" punch in column 13 will cause values punched in the first four card fields to read simultaneously and directly into the 412 counters 71, 72, 73, 74. This is called <u>spread entry loading</u> or, less formally, <u>spread reading</u>.

5.3.6.1 Spread read quantities may be addressed immediately on the next card.

5.3.6.2 Results may not be stored in counters 71-74 on a card immediately preceding a spread entry card. (See also 9.1.4)

5.3.6.3 Addresses 71-74 and 81-84 may not be used in U, V, and W on a card with a spread entry.

5.4 Three addresses are used to recall information from electronic storage.

5.4.1 49: Read the preceding Z result into any of U, V, and W.

5.4.2 39: Read the preceding Y result into any of U, V, and W.

5.4.3 99, for V, and/or W: Use the values for V and/or W which were used in the previous card cycle. (See also 6.2.4)

5.4.4 The electronic storage registers in which the U, V, and W operands are placed are reset automatically every card cycle except when the retention code 99 is used.

5.5 09: Read the U, V, and/or W values from the corresponding card fie...
(Fields 5, 6, and/or 7.)

5.5.1 The card field takes precedence over any other address. The... punched in fields 5, 6, or 7 will disturb any read in other than 09.

5.5.2 A blank in the W address behaves as 09. U and V addresses must be coded. (See also 7.1)

5.6 Table of Sources which each operand register may accept:

Types of Sources	Register U	V	W	Address Code
Card fields 5, 6, and/or 7.	F5	F6	F7	09
Any 2 numbers in the 412 or 941 storage.	S	S* or S*		11-18; 21-28; 71-77
Preceding Y result.	Y	Y	Y	39
Preceding Z result.	Z	Z	Z	49
V and/or W operand of preceding (retained in electronic registers).	-	V	W	99

5.6.1 *When one of these two operands is supplied from mechanical storage, the other must be read from the card, be retained unaltered from the previous line, or be obtained from one of the two previous results. Both may not come from mechanical storage. (See 3.4.5) (See also 5.8)

5.7 Certain of these addresses may be modified in several ways.

5.7.1 An "X" punch in the units position of all addresses except 99 serves to reverse the sign of U, V, and W operands.

5.7.2 To obtain a positive absolute value of a quantity, read the quantity from storage with its sign reversed and punch a "9" (minus sign) in the sign column of the corresponding U, V, or W field of the card. To obtain the negative absolute value of a quantity, read that quantity from storage into U, V, or W and punch a "9" (minus sign) in the sign column of the corresponding U, V, or W field of the card.

5.7.3 A "Y" punch in the 10's position of all addresses except 99 will shift U, V, and W one place to the left, effectively multiplying that operand by 10 before entry into the electronic registers for computation.

5.8 To facilitate programming, provision has been made for permutation of operands after they are read in but before they are computed on.

5.8.1 Table of Permutations

Type of Action	Contents of registers after operands have been permuted, but before state of arithmetic operations			Code Column 13
none	U	V	W	none
U → V, V → U	V	U	W	1
U → V, V → W, W → U	W	U	V	2
U → W, W → U	W	V	U	3
U → W, V → U, W → V	V	W	U	4
V → W, W → V	U	W	V	5

5.8.2 Contents of registers remain in their permuted arrangement after the arithmetic operations have been performed.

5.8.3 On the program sheet it is customary to write the algebra in the permuted arrangement, although the addresses, of course, are written and punched in the read in arrangement.

5.9 Special Storage Reset Codes

5.9.1 To reset the 941 storage, two cards are needed for any one bank:

5.9.1.1 bank 1 (storages 11-18)

Card	Code Column 22
1.	1
2.	X

5.9.1.2 bank 2 (storages 21-28)

Card	Code Column 22
1	2
2	X

5.9.1.3 These four cards can be condensed to three. (See 5.9.3)

5.9.2 An "X" punch in column 19 is used to reset all the 412 counters simultaneously.

5.9.2.1 It is permissible to clear the 412 storage with an "X" punch in column 19 and on the same card store a result in the 412 via channel "C."

5.9.3 All 941 storage and 412 counter units are reset before starting a calculation, or at any place in the sequence, by using three cards:

Card	Code Column 19	Code Column 22
1		1
2		X2
3	X	X

6. Algebraic Operations

6.1 The CPC Model II performs two algebraic operations with three operands on each card.

6.1.1 The first operand, called U, can be combined with the second operand, V, to produce the first result, Y. Y may be combined with the third operand, W, to produce the second result, Z. Either Y or Z may be stored and printed (via channel "C"), and/or summary punched; both may be retained as operands in the next computing step.

6.1.2 The first operation code is a single digit which is punched in column 16. The second operation code may consist of 1, 2, or 3 digits, all punched in column 19.

6.2 Table of Algebraic Operation Codes

Name of operation	Symbol	Code Column 16	Code Column 19
Transfer U to Y	↑	1	
Add (algebraically)	+	2	2
Divide (normal)	/	3	3
Divide (inverted)	\	4	4
Multiply	·	5	5
Extract square root of "W"	√		6

6.2.1 To subtract reverse the sign of the operand and add, using operation code (2).

6.2.2 Two divisions (whether direct (3) or inverse (4)) cannot be performed on the same card cycle.

6.2.3 A division may not be performed on the same line with (6), the square root of W. (See 6.2.3.1 and 6.3.4.2)

6.2.3.1 When the second operation calls for the square root of W (6) and the first operation is coded as division (3), the dividend will be the number in register W (rather than U).

6.2.4 The operand in register V cannot be retained for use on the next line of programming if the second operation is a square root. It is replaced by the final value of the square root.

6.2.5 Correct performance requires that two operations be always coded on a card.

6.3 It is possible by the use of the following substitution codes to vary the operands of the second operation.

6.3.1 Table of Substitution Codes

Name of operation	Symbol	Code in Column 19
*Substitute V for Y	V	7
*Substitute Y for W	Y	8

6.3.2 *These substitutions do not affect the contents of the V, Y, or W registers but only the way in which these operands are associated during the second operation in a card cycle.

6.3.3 In a Y for W substitution (8), a W addressed operand is read in but the operation is entirely independent of it.

6.3.4 Table of Permissible Associations of Operands

1st Operation	2nd operation	Substitution Code Column 19
$U \div V = Y$	$Y \cdot W = Z$	(normal mode)
	$V \cdot W = Z$	7
	$Y \cdot Y = Z$	8
	$V \circ Y = Z$	7, 8
$U \circ V = Y$	$\sqrt{W} = Z$	(normal mode)
	$\sqrt{Y} = Z$	8

6.3.4.1 No substitutions are possible in the first operation.

6.3.4.2 When the second operation is the square root of Y (6,8) the first operation may be the division of U by V (3).

6.4 A second variation in the algebraic operations is the shift right of results Y and/or Z, effectively multiplying by a positive integral power of 1/10 and rounding.

6.4.1 Table of Shift Right of Results

Type of Action	Column	Code
Shift Y 1 place to right ($x10^{-1}$)	13	9
Shift Y 2 places to right ($x10^{-2}$)	13	8
Shift Y 3 places to right ($x10^{-3}$)	13	7
Shift Z 1 place to right ($x10^{-1}$)	24	9
Shift Z 2 places to right ($x10^{-2}$)	24	8
Shift Z 3 places to right ($x10^{-3}$)	24	7

6.4.2 The shift takes place immediately as a part of the first or second operation. In the case of Y, the shifted Y then enters the second operation.

6.5 The following properties of the machine should be noted.

6.5.1 The arithmetic unit interprets a sequence of eight decimal digits as a number in decimal notation with the decimal point occurring immediately after the first digit, i.e., XXXXXXXX is interpreted as X.XXXXXXX.

6.5.1.1 Numbers going into and out of the electromechanical storages must be regarded in this way also. One exception is storage 71, which has no units position and must therefore be treated as ± 0.XXXXXXX. There is a plug on the piggy-back portion of the control panel by which the sign control of this storage may be disconnected. With the sign control off, 71 may be used for storage of positive numbers only, + 0.XXXXXXX, and column 3 may be used as an additional column for card identification. With the sign control on, any non zero digit punched in column three will cause the contents of 71 to be negative.

6.5.2 If any result Y or Z exceeds 9.9999999 but is less than 100, the first digit on the left will normally be truncated. However, one and only one such overflow digit on the left may be salvaged by using the shift result one place right with this operation.

6.5.3 If a quotient exceeds or equals 100, the machine gives 0 as a result.

6.5.4 Division by 0 gives 0.

6.5.5 The square root of a negative number will be zero.

7. Transcendental Functions

7.1 For computing transcendental functions the argument is addressed as a U operand, and the two-digit operation code is placed in columns 16 and 19. V and W addresses must be left blank. Circular and hyperbolic functions are generated for radian arguments only.

7.1.1 The 99 code may not be used following a card on which a transcendental function is called for.

7.1.2 The result of a transcendental function calculation is called from electronic storage on the next line by a 49 code.

7.2 The control panels are wired to calculate the functions at electronic speed recursively by the following series:

7.2.1 $\cos u = 1 - \frac{u^2}{2!} + \frac{u^4}{4!} - \frac{u^6}{6!} + \ldots$

7.2.2 $\cosh u = 1 + \frac{u^2}{2!} + \frac{u^4}{4!} + \frac{u^6}{6!} + \ldots$

7.2.3 $\sin u = u - \frac{u^3}{3!} + \frac{u^5}{5!} - \frac{u^7}{7!} + \ldots$

7.2.4 $\sinh u = u + \frac{u^3}{3!} + \frac{u^5}{5!} + \frac{u^7}{7!} + \ldots$

7.2.5 $e^u = 1 + u + \frac{u^2}{2!} + \frac{u^3}{3!} + \frac{u^4}{4!} + \ldots$

7.2.6 $\arctan u = u - \frac{1}{3} u^3 + \frac{1}{5} u^5 - \frac{1}{7} u^7 + \ldots$

7.2.7 $\operatorname{arctanh} u = u + \frac{u^3}{3} + \frac{u^5}{5} + \frac{u^7}{7} + \ldots$

7.2.8 $\ln u = (u - 1) - \frac{1}{2} (u - 1)^2 + \frac{1}{3} (u - 1)^3 \quad \ldots$

7.2.9 The limits on the arguments are determined by:

1. Rate of convergence of the series.
2. Size of the resulting function.
3. Size of the intermediate calculations of the series.

7.3 Table of Transcendental Function Codes

Function	Code Column 16	Code Column 19	Allowable Argument Range	Remarks
cos	8	2	$-\sqrt{10} < u < \sqrt{10}$	range restricted by u^2 *.
sin	8	3	$-\sqrt{10} < u < \sqrt{10}$	" " " " *.
cosh	9	2	$-\cosh^{-1}10 < u < \cosh^{-1}10$	range restricted by value of fctn.; $\cosh^{-1} 10 = 2.9982$
sinh	9	3	$-\sinh^{-1}10 < u < \sinh^{-1}10$	range restricted by value of fctn.; $\sinh^{-1} 10 = 2.9932$
e^x	7	5	$(-\sqrt[3]{60} < u < \ln_e 10)$	lower range restricted by $u^3/3!10$;** upper range restricted by value of fctn.; $\ln_e 10 = 2.3026$ exp $2.3025851 = 10$
$\tan^{-1}$	8	7	$-.91 < u < .87$	range restricted by desired accuracy***.
$\tanh^{-1}$	9	7	$-.87 < u < .87$	range restricted by desired accuracy***.
$\ln_e$	7	9	$(0.4 < u < 1.9)$	range restricted by desired accuracy.

7.3.1 The error in the computed function for arguments within the above range is less than 10^{-6}.

7.3.2 *u^2 is stored in the electronic computing unit (605) in the course of this calculation.

**This is the fourth term in the series representation of e^u used.

***Terms of series are summed until the exponent of the argument reaches 100, or until a term of the series is less than 5×10^{-8}, whichever occurs first.

7.4 Subroutines for extending ranges of arguments of these functions may be constructed from the following relations.

7.4.1 $\sin x = -\sin(x - 2\pi n - \pi)$ n an integer and
$\cos x = -\cos(x - 2\pi n - \pi)$

7.4.2 $\arctan x = 2 \arctan\left(x / 1 + \sqrt{1+x^2}\right)$, $-\frac{\pi}{2} \leq \arctan x \leq \frac{\pi}{2}$

7.4.3 $\arcsin x = \arctan x / \sqrt{1-x^2}$, $-\frac{\pi}{2} \le \arcsin x \le \frac{\pi}{2}$

$\arccos|x| = \arctan \sqrt{1-x^2} / |x|$, $0 \le \arccos x \le \frac{\pi}{2}$

7.4.4 $\text{arctanh } x = 2 \arctan x / (1 + \sqrt{1+x^2})$

7.4.5 $\text{arcsinh } x = \text{arctanh } x / \sqrt{1+x^2}$

$\text{arccosh } x = \text{arctanh } (x^2-1) / x$

7.4.6 $\ln x = 2 \text{ arctanh } (x-1)/(x+1)$

$\ln x = n \ln \left(x^{\frac{1}{n}}\right)$

7.5 Let $\arctan (y / x) = \varphi$, $0^\circ \le \varphi < 360^\circ$;

let $\arctan (y / x) = \theta$, $-\frac{\pi}{2} \le \theta \le \frac{\pi}{2}$, be the angle given by the machine;

let $\epsilon = |y| / y$, $\delta = |x| / x$, $(\epsilon, \delta = \pm 1 \sim 0)$

let $10^{-2} \omega = \left[\frac{1-\delta}{2} + \frac{1+\delta}{2} (\epsilon - 1) \epsilon \right] 1.80$,

then $10^{-2} \varphi = \left(\frac{1.80}{\pi} \theta + 10^{-2} \omega \right) \frac{x}{x} + \left(1 - \frac{x}{x}\right)(1.8 - .9 \epsilon)$

8. Branch Orders

8.1 It is possible to test, and in effect store, the sign of the result of a calculation and on the basis of this test obey certain subsequent instructions and ignore others.

8.2 At any instant one and only one sign is remembered by the machine. This sign is remembered up to and including a card with a test instruction.

8.3 A zero is interpreted by the machine as a positive number.

8.4 A card with an "8" or a "9" code in column 26 is called a <u>conditioned card</u>. A card which is not conditioned is called an <u>unconditioned card</u>. A card is called a suppressed card if it is conditioned by an "8" and the remembered sign is positive, or if it is conditioned by a "9" and the remembered sign is negative; this depends on the history of the machine preceding the reading of the card. A card is called an <u>unsuppressed card</u> if it is an unconditioned card or if it is a conditioned but not a suppressed card.

8.5 The instructions of unsuppressed cards are obeyed.

8.6 The computed results, (standing in the Y and Z registers), of a particular card are not disturbed when followed by suppressed cards. Hence these results may be called back by 39 and 49 codes in the address positions of the first unsuppressed card following this particular card and on all intervening cards.

8.7 On suppressed cards orders to read into U, V, and W, to clear the 412 storage unit, to clear the 941 unit, to eject, to spread read or to spread read print, (see 9), are obeyed. All other orders are ignored.

8.8 Table of Branch Orders

Action	Code in Column 26
8.8.1 Inspect and remember the sign of the Y result of this card.	6
8.8.2 Inspect and remember the sign of the Z result of this card.	7
8.8.3 Suppress computing, storing, punching, plotting and the action of an "X" punch in column 26, (see 9.1.1 and 9.4), <u>if remembered sign is positive</u>.	8
8.8.4 Suppress computing, storing, punching, plotting and the action of an "X" punch in column 26, (see 9.1.1 and 9.4), <u>if remembered sign is negative</u>.	9
8.8.5* Stop card feed and suppress as in 8.8.3 <u>if remembered sign is positive</u>.	5, 8
8.8.6* Stop card feed and suppress as in 8.8.4 <u>if remembered sign is negative</u>.	5, 9
8.8.7 Suppress test of the Y result of this card and suppress as in 8.8.3 <u>if remembered sign is positive</u>.	6, 8
8.8.8 Suppress test of the Z result of this card and suppress as in 8.8.3 <u>if remembered sign is positive</u>.	7, 8
8.8.9 Change remembered sign to positive and suppress as in 8.8.4 <u>if remembered sign is negative</u>.	6, 9
8.8.10 Change remembered sign to positive and suppress as in 8.8.4 <u>if remembered sign is negative</u>.	7, 9

8.8.11 *Card feed will stop when second card following <u>stop card</u> enters the feed.

9. Printing

9.1 Two kinds of printing are possible: printing of a result, which occurs a card cycle after the instruction to print, and spread-read printing, which occurs on the same card cycle as the instruction to print.

9.1.1 An "X" punch in column 26 together with a <u>print field</u> punch "1" - "8" in column [illegible] causes one result Y or Z, whichever is on channel "C," to print in one of 8 positions or print fields across the page.

An "X" punch in column 26 also causes quantities which are punched in card fields 1-4, if any, on the following card to print in print fields 1-4.

9.1.2 An "X" punch in column 25 causes quantities punched in card fields 1-4 on the same card to print simultaneously in print field 1-4. This is called spread read printing. The quantities need not be spread read into storage. Blank columns will not print as zero.

9.1.3 An "X" punch in column 26 followed on next card by an "X" punch in column 25 will cause type bars to rise only once.

9.1.4 A result to be printed must not be directed to print in any of the print fields 1-4 if there are punches in the corresponding card fields of the following card, or if the following card is a spread entry card.

9.1.4.1 In this situation the machine would print and/or read into storage the higher digit in every position.

9.1.5 Y or Z need not be stored with printing. However, it is not possible to print one and store the other.

9.1.5.1 The final card in the 412 card feed will not print a result.

9.1.6 Table of Print Codes

Print in field	Code: Column 25	Code: Column 26
1	1	X
2	2	X
:	:	
8	8	X
1-4	X	

9.2 An "X" punched in column 24 causes suppression of printing from the type bars where long hammerlocks are raised when printing is ordered on the same card by "X" in 25 or on preceding card by "X" in 26.

9.3 A result Y or Z may be shifted from 1 to 5 places to the right to align the printed values around a fixed decimal point. This coding causes truncation of the printed result without rounding, but does not affect storage of the result, nor output punching, nor subsequent calculations using this result. This shift takes place after and independently of the right shift described in section 6.4.

9.3.1 Table of Right Printing Shifts

Action	Code Column 24
Shift printed result 1 place to right	1
Shift printed result 2 places to right	2
Shift printed result 3 places to right	3
Shift printed result 4 places to right	4
Shift printed result 5 places to right	5

9.4 A "1," "2," or "3" punch in column 26 with an "X" punch in column 26 on the same card or an "X" punch in column 25 on the following card causes the paper to space 1, 2, or 3 lines before printing.

9.4.1 The above codes are suppressed when printing is suppressed by branch order.

9.4.2 A "4" punch in 26 will cause paper to eject to a new page with or without printing.

9.4.3 Spacing and ejecting take place before printing.

9.4.4 Spacing 1-3 lines will not take place on first print order after card feed of the 412 unit has been cleared of cards.

9.4.5 An "X" in 26 with no field punch in 25 will cause type bars to rise, spacing may take place, and fields 1-4 of next card may print, but result will not print.

9.4.6 Table of List Spacing Codes

Action	Code Column 25	Code Column 26	
Space 1 line		X1	Card 1
	X		Card 2
Space 2 lines		X2	Card 1
	X		Card 2
Space 3 lines		X3	Card 1
	X		Card 2
Eject page		4	with or without X in 25 or 26

9.5 Remarks on the 412 as a Printer

9.5.1 Fields 2-7 may print 8 digits, a decimal point, and a sign at the right of the field.

Field one may print 7 digits and a sign at the left of the field. The sign of field one does not print on set up.

Field 8 may print 8 digits and a sign, or 7 digits, decimal point, and a sign.

9.5.2 In fields 2-4 the decimal must remain fixed in the second printing position, but the digits may be displaced to the right or left by moving the plugs on the piggy back portion of the 412 control panel. In fields 5-8 the position of the decimal may be changed by moving the plugs on the piggy back.

9.5.3 Long or short hammerlocks may be raised over any individual type bar to prevent printing in that position. Short hammerlocks prevent all printing. Long hammerlocks are controlled by a coded instruction.

9.5.4 Zero suppression levers may be raised to suppress printing of zeros to the right within a field and may be linked to force zeros to print to the left.

10. Plotting

10.1 A "Y" punch in column 26 will cause plotting of a point whose x and y coordinates are stored in 412 storage 76. The first four digits represent x and the last four digits y.

10.1.1 It is not possible to use both plotter and summary punch on the same run.

10.1.2 The "Y" punch may be on first card after storing in 76 or later.

10.1.3 A blank card (i.e. no instructions) must follow each card which orders plotting.

10.1.4 Only positive coordinates may be plotted.

10.1.5 Four scale factors are available for either axis independently:

Scale	factor	1	means head moves 1 space	per count of the number being plotted
"	"	2	"	per 2 counts
"	"	4	"	per 4 counts
"	"	5	"	per 4 counts

10.1.6 The origin may be selected on either axis independently by setting it manually into the four x and the four y reset selector switches.

10.1.7 Plotted points must be at least 2 cm. from margins of paper.

10.1.8 The maximum plotting area of the table top is 26 inches by 54 inches.

10.2 Consideration must be given to scaling the coordinates of points to be plotted in order to obtain a graph of the desired dimensions.

10.2.1 We shall interpret the sequence of four digits representing x as an integer. We introduce the following notation:

x_e is largest value

x_f is smallest

$x_e - x_f = r_x$ (range of x)

10.2.2 Select a suitable multiple of 100 just smaller than x_f. Call it x_o and subtract it from all x values. Call these $\bar{x}$.

10.2.3 The smallest division on the paper is a millimeter. Decide how many millimeters the graph is to cover. Call this d. (N.B. Maximum d on vertical scale is 460 mm. for the stock paper. The horizontal scale may cover 1350 mm.)

10.2.4 Divide $r_x + x_f - x_o$ by d. Call this k.

10.2.5 If k > 0.5, scale factor is 1, 2, 4, or 5, whichever is next greater than k. If k is not near the scale factor, the graph will extend only partly over the region.

10.2.6 If k < 0.5, scale factor is 1. If it is desired to enlarge the graph, calculate 1/k and program a multiplication of all $\bar{x}$ by 2, 5, or 10, whichever is next smaller than 1/k. Any other factors must be used with care since they would make a difficult scale to read and might exaggerate the significance of the data.

10.2.7 Store $\bar{x}$ in first four positions of 76 if it is the horizontal coordinate, last four positions of 76 if it is the vertical coordinate.

10.2.8 At the plotting origin, $x = x_o$. It is convenient to have this 2 cm. from bottom of page and chosen so that the true zero would be on a heavy line.

10.2.9 Instead of subtracting x_o from x and working with $\bar{x}$, it is possible to set x_o into the reset selector switches and work with x directly. But it will still be necessary for x to be always positive.

10.2.10 It is possible to plot several curves at one time if a convenient constant can be chosen to displace the curves and make them distinct.

11. Summary Punching

11.1 A "Y" punched in column 26 activates the 527 output punch. A summary punch control panel must be wired and inserted into position.

11.1.1 Even if no punching is to be done, a punch control panel must be placed in the 527 with the calculate switch wired on.

11.2 The contents of any of the 412 storages and of the 604 electronic storages for Y, Z, and W can be punched on the same cycle.

11.2.1 Punching from the 412 storage may proceed on the next card after addressing a result to the 412.

11.3 The alignment of punched digits in the fields of the output cards is not affected by the shift codes of sections 5.7 and 9.3, but it is affected by the shift codes of section 6.4. Output punching is controlled by branch order codes in the same way that printing and storing is controlled.

IV PLANNING AND CHECKING THE PROGRAM

12. Identification

As a preliminary, the programmer should make a point of studying the problem, breaking it into phases of computation each of which will be accommodated by the available storage. He should plan in detail the provisional format of printed results desired. Along with this effort the planning of the format of the input data card is to be accomplished. This format must correspond to the coding on the program sheet; hence in preparing such sheets, it is important to plan the pattern of identification of the cards.

Identification consists in an arrangement of the parameters, variables, and cycles which describe each main part of the problem, and is contained chiefly in field 1 of the cards, although it may be included in fields 2, 3, and 4, if necessary.

Generally in this system, each card will contain an instruction and will control two arithmetical operations. An elementary set of cards will represent a complete sequence of operations sufficient for computing all functions of one set of parameters and variables. The same set of cards can be duplicated automatically so that as many sets may be used to compute the functions as there are unique sets of values of parameters and variables in the problem. Thus there will be as many elementary sets of instruction cards or kernels as there are combinations of change in parameters and variables. A data card or a special set of instruction cards is used to change these values, and are usually inserted ahead of each standard set of instructions. Thus in between each kernel, a few of these special cards are inserted.

Insertion can be carried out automatically on the collator only if identification is systematic, e.g., sequential or lexicographic in nature. In general the identification of a data card should correspond with that of the instruction cards of a kernel. Exceptions are, however, possible. Data or special cards are usually distinguished from kernel cards; this distinction is adequately provided by the assignment of distinctive line numbers (in card columns 11 and 12) to each class of card; e.g., data cards may have low-valued line numbers and kernel cards high-valued line numbers. Generally, each line number corresponds to a line on the program sheet.

Provision for proper identification in the above fields in the cards should take into account the peculiar identification requirements of the printed format of the final results. Thus each line and section of printed results may need several items of identification. If there are to be several pages of results, provision should be made for identifying at the top or bottom to every page the value of every parameter and variable applicable to that page. Provision likewise should be made for printing the MAR (project number) or case (e.g., ARR number) on every page. This may be done by proper insertion in the card file of special data or instruction cards between every twenty or so kernels. Often the cards required to produce one page of results will be sufficient for the entire problem, for other pages can be produced with only a change in the leading cards.

13. Program Tracing

13.1 In starting a problem on the CPC one or two hand-computed or one or two special automatically computed pilot problems should be used as test cases. If there are differences between expected results and these test cases, program tracing can be used to detect the area of the program where trouble occurs. Once these cases are "checked out," a set of cards from the standard program should be kept on hand for future testing of machine reliability for that problem, and a program tracing of the set of cards should be kept for ready reference. There also should be prepared for ready reference a typical printed set or page of final results.

13.2 To trace program and machine sequences a switch can be thrown causing printing of each instruction code, corresponding operands and one result. This is called listing or setting up. It is advisable to check a program by this method before all of the kernels are reproduced.

13.2.1 Columns 3-26 inclusive print on the left half of the page. On the right half of the page, the operands U, V, and W will print in print fields 5-7 and the Y or Z result, whichever is on channel "C," in field 8. Fields 5-7 will print 7 digits to the right of the decimal. Field 8 will print 8 digits without a decimal. A minus sign will print at the right side of these fields.

13.2.2 Coded sign reversals ("X" in units position of the address codes) will print on the letter R to the right of the address code.

13.2.3 The value retained by a 99 code will not print.

13.2.4 Minus signs are printed to the right of the number.

13.2.5 Lines which contain permutation and/or substitution codes print on set up with operands in read in locations.

13.2.6 An "X" in column 19 to clear the entire 41? storage will print C at the extreme left.

13.2.7 A spread read card will print E at the extreme left.

13.2.8 Column 3 will not print on set up.

13.2.9 A positive absolute value is indicated by a P to the right of the address code. A negative absolute value is indicated by an N to the right of the address code.

14. Checking and Roll Back

The machine must not be relied on to produce error-free computation, nor is there any greater probability that its errors will tend to be small instead of large, obvious instead of insidious or self-converting rather than cumulative. Therefore in general all problems should be run twice (preferably on different machines). However, when there are obvious physical criteria for judging results on a precise basis, or frequent analytic checks that may be programmed systematically, reruns may not be necessary. An exception of another sort is met when an iterative procedure is used, wherein small errors will be erased by the iterative process. In this connection large errors may lead to convergence on another branch which may or may not seem plausible, so that the nature of the convergence process requires analysis or experiment for each problem. Another exception, where rerunning may not be necessary, occurs where an index of improvement between coarse and fine integrations is computed and printed as a visual aid for quick trouble-spotting. Smoothness checks for certain problems may be adequate, but cannot be relied on where functions of more than one variable are computed, or where the differences which are not used for trouble spotting are too large or too variable inherently.

In general, all automatically computed results should be transmitted from the Applied Mathematics Laboratory only after complete checking has been carried out.

Most computing equipment errors are intermittent and will not repeat during a tracing program, so that quick electronic or mechanical repair cannot be carried out. Under these somewhat typical circumstances, program tracing is unprofitable, and it becomes necessary to continue computation of the problem at the last point where results are known to be perfect. To commence automatic computation "in the middle" of a large problem for this purpose, a so-called <u>rollback</u> procedure must be programmed. Such a program is designed to store all information necessary to resume the calculation in the particular storages that are used in the standard program for that problem. The items of this information may be available in basic data cards used in the beginning of the computation, or in the middle, or they may have been computed and printed, or they may be summary punched. To insure that all information which is necessary to rollback a problem be available in card form in readiness for use, the mathematician must foresee where a rollback will start (say at the beginning of any printed page) and then plan his standard program in such a way as to compute and summary punch or print each item of such information. Of course, any such items which exist on the standard data and instruction decks need not be re-punched.

Once the necessary items are available, a standard rollback program and deck of cards is prepared and checked out, which will assign the information to storage locations which are standard in the problem program; likewise this rollback kernel will compute all quantities derived from this basic information and which are stored in standard places in the problem program. In short, a rollback kernel includes all data and instruction codes necessary to provide a configuration of storage standard to the problem program, such that a standard kernel of instruction cards immediately following it can be used to continue the computation in normal fashion. When a rollback kernel is checked out, a program tracing should be made for future reference, and a concise and complete set of directions for assembling the rollback kernel should be written out.

15. Outline of Programming Steps

The programming steps, aside from problem formulation and numerical analysis, includes the following.

15.1 Translation of the problem which is stated in discrete terms into sequences of algebraic steps, etc.

15.2 Formulation of a logistical pattern of storage, etc.

15.3 Formulation of a related and useful pattern of presentation of desired results.

15.4 Formulation of a rollback procedure wherein all necessary cards for special data are produced automatically by summary punching.

15.5 Coding the above, in standard forms, on program sheets, and punching corresponding cards.

15.6 Establishing the data card pattern and transmitting this information to the problem proposers so they can furnish appropriately punched input cards.

15.7 Pilot computation, and test decks construction; program check-out and establishment of standard program trace for reference.

15.8 Establishment of standard printed page of final results for diagnostic reference; this should include identification interpretation, and column heading interpretation, etc.

15.9 Preparing summary card format, in written form, and summary control panel, wherever summary punching is used.

15.10 Preparing all control panels which may be needed in preparing the deck of cards to be used in the Card-Programmed Calculator, with specifications for their preparation and use.

15.11 When a problem is recurrent or of large volume, preparing explicit notes on all the above results which are sufficient guides to allow processing by any other programming or machine technician. These notes should be created first in the normal course of any programming procedure, for even the original programmer can easily forget important details.

15.12 Operating, checking, maintaining liaison, and taking and transmitting final results to the user (problem proposer). In these phases, maintenance of the CPC log books should be kept consecutively and dates of transmittal of each portion of results should be recorded on facsimile bound copies retained in AM II. All transmitted results and these AM II copies should be initialed by the programmer or machine technician who has carried out operation for the bulk of the book.

15.13 When a problem has been completed, an AML report of completion (where AML stands for "Applied Mathematics Laboratory") should be immediately prepared and transmitted to AM, showing briefly what has been completed; this is an administrative rather than a technical report. Furthermore, where a problem has been partially or wholly formulated in the branch or division, or where novel application and development of numerical analysis has been made, a formal Technical Note or NavOrd Report should be prepared as soon as possible.

APPENDIX

16. The foregoing coding instructions may be applied to the CPC Model I with the exceptions and additions noted below.

16.1 The following are not available on Model 1.

Shift U, V, W (5.7.3)
Shift Y, Z (6.4)
Read 49 into V register
Read 39 into W register

16.1.1 Both banks of the 941 electromechanical storage are reset to zero by an "X" punch in column 20.

16.1.2 Any number stored in 412 storage 71 must be less than 0.1000000.

16.1.3 The absolute value, positive or negative, of the previous Z result may not be called into W.

16.2 On the CPC Model I, transcendental functions are available only when special function panels are used. These panels cause the Model I to operate as a single operation, two address system for algebraic operations; single operation, single address system for square root and transcendental functions.

16.2.1 The W address is always blank.

16.2.2 For square root and transcendental functions the argument is addressed to U and the V address is 09.

16.2.3 Table of Function Board Codes

Operation	Code Column 16	Code Column 19
U plus V	2	0
U plus V plus previous result	2	2
U divided by V	3	0
V divided by U	4	0
Remainder of U divided by V after 5 places of quotient	3	5
Remainder of V divided by U after 5 places of quotient	4	5
U multiplied by V	5	0
U multiplied by V plus previous result	5	2
cosine U	2	8
hyperbolic cosine U	2	9
sine U	3	8
hyperbolic sine U	3	9
exponential U	5	7
square root U	6	7
arc tangent U	7	8
arc hyperbolic tangent U	7	9
natural logarithm U	9	7

APPENDIX

continued

16.2.3.1 The previous result is addressed as 49.

16.2.3.2 There is no 99 address with this panel.

16.2.3.3 The range of permissible arguments is same as for Model II.

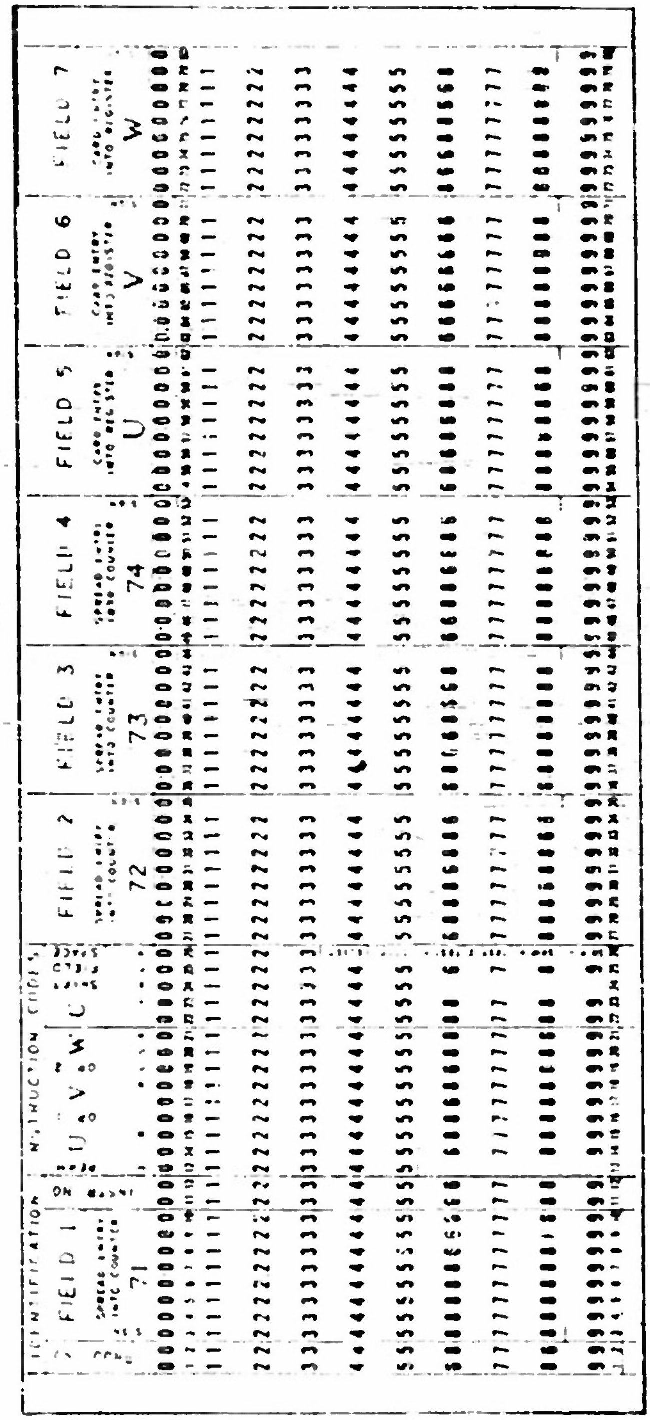
16.2.4 Section 13.2.9 does not apply to the Model I. A positive absolute value is indicated by an R to the right of the address code. A negative absolute value is not indicated.

Attachment 1

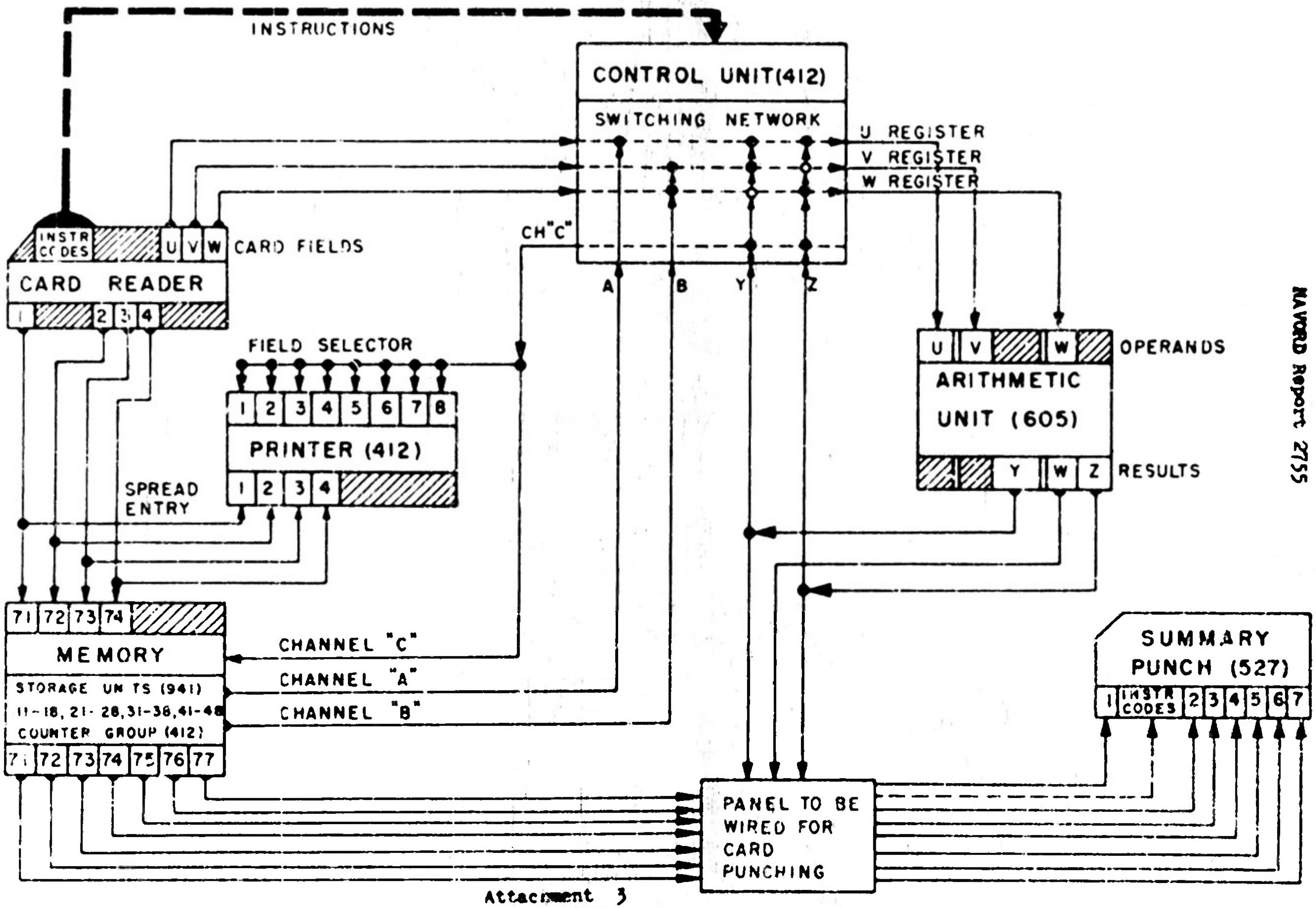
Card No.	Perm.	Register U	1st Op.	Register V	1st Result = Y	2nd Op.	Register W	2nd Result = Z	Ch. C	Shift	Field	Sp. Op.
7 8 9 10 11 12	13	14 15	16	17 18		19	20 21		22 23	24	25	26

DOUBLE OPERATION PLANNING CHART FOR CARD PROGRAMMED CALCULATOR

Attachment 2



BLOCK DIAGRAM FOR CPC GENERAL-PURPOSE CONTROL PANEL



Aeroballistic Research Department
External Distribution List for Applied Math (X 3)

No. of Copies	
	Chief, Bureau of Ordnance
	Department of the Navy
	Washington 25, D. C.
1	Attn: Rea
1	Attn: Rexe
3	Attn: Re9a
	Chief, Bureau of Aeronautics
	Department of the Navy
	Washington 25, D. C.
1	Attn: AER-TD-414
1	Attn: RS-7
	Commander
	U.S. Naval Ordnance Test Station
	Inyokern
	P.O. China Lake, California
2	Attn: Technical Library
1	Attn: Code 5003
	Commander
	U.S. Naval Air Missile Test Center
	Point Mugu, California
2	Attn: Technical Library
	Superintendent
	U.S. Naval Postgraduate School
	Monterey, California
1	Attn: Librarian
	Commanding Officer and Director
	David Taylor Model Basin
	Washington 7, D. C.
1	Attn: Hydrodynamics Laboratory
	Chief of Naval Research
	Navy Research Section
	Library of Congress
2	Washington 25, D. C.
	Office of Naval Research
	Washington 25, D. C.
1	Attn: Code 432
1	Attn: Code 438
1	Attn: Code 463
	Director
	Naval Research Laboratory
	Washington 25, D. C.
1	Attn: Code 2021
1	Attn: Code 3800
	Research and Development Board
	Library Branch
	Pentagon, 3D1041
1	Washington 25, D. C.
	Chief, AFSWP
	P.O. Box 2610
	Washington, D. C.
1	Attn: Technical Library

No. of Copies	
	Air Targets Division
	Directorate of Intelligence
	Headquarters, USAF
	Washington 25, D. C.
1	Attn: Chief, Physical Vulnerability Br.
1	Attn: AFDRD-RE
	Commanding General
	Hdqts. Wright Air Development Center
	Wright-Patterson Air Force Base
	Dayton, Ohio
1	Attn: WCSO
7	Attn: WCAPD
2	Attn: WCRR
	Physics Branch
	Division of Research
	U. S. Atomic Energy Commission
1	Washington 25, D. C.
	Director
	Maxwell Air Force Base, Alabama
1	Attn: Air University Library
	The Artillery School
	Antiaircraft and Guided Missiles Br.
	Fort Bliss, Texas
2	Attn: Research and Analysis Sec.
1	Attn: Pres. Army Field Forces, Board #4
	Ballistic Research Laboratories
	Aberdeen Proving Ground
	Aberdeen, Maryland
2	Attn: D. S. Dederick
	National Bureau of Standards
	Washington 25, D. C.
1	Attn: Nat'l Applied Math. Lab.
1	Attn. W. Ramberg
1	Attn: Librarian (Ord. Dev. Div.)
	National Bureau of Standards
	Corona Laboratories (Ord. Dev. Div.)
	Corona, California
1	Attn. Dr. Harold Thomas
	National Bureau of Standards
	Building 3U, UCLA Campus
	405 Hilgard Avenue
1	Los Angeles 24, California
	Attn: Librarian
	University of California
	Berkeley 4, California
1	Attn: G. J. Maslach
1	Attn: Dr. S.A. Schaaf
	VIA: InsMat
	California Institute of Technology
	Pasadena 4, California
2	Attn: Librarians (Guggenheim Aero. Lab.)
1	Attn: Prof. M.S. Plesset
	VIA: BuAero Rep.

No. of Copies

University of Illinois
202 E. R. L.
Urbana, Illinois
1 Attn: Prof. A.H. Taub
VIA: InsMat

Director
Fluid Dynamics and Applied Math.
University of Maryland
1 College Park, Maryland
VIA: InsMat

University of Michigan
Engineering Research Institute
Ypsilanti, Michigan
1 Attn: J.W. Carr, III

Willow Run Research Center
Willow Run Airport
University of Michigan
Ypsilanti, Michigan
1 Attn: L.R. Biasell
VIA: InsMat

University of Minnesota
Rosemount, Minnesota
1 Attn: J. Leonard Frame
VIA: Ass't InsMat

Dept. of Aero. Engr.
The Ohio State University
Columbus, Ohio
1 Attn: G. L. Von Eschen
VIA: Ass't InsMat

Polytechnic Institute of Brooklyn
99 Livingston Street
Brooklyn 2, New York
1 Attn: Dr. Antonio Ferri
VIA: ONR

University of Wisconsin
Naval Research Laboratory
422 Farm Place
1 Madison 6, Wisconsin
VIA: InsMat

Princeton University
Forrestal Research Center Library
Project Squid
Princeton, New Jersey
1 Attn: M. H. Smith

Armour Research Foundation
35 West 33rd Street
Chicago 16, Illinois
1 Attn: Engr. Mech. Div.

Applied Physics Laboratory
The Johns Hopkins University
8621 Georgia Avenue
Silver Spring, Maryland
1 Attn: Arthur G. Norris
VIA: NIO

No. of Copies

Defense Research Laboratory
University of Texas
Box 1, University Station
1 Austin, Texas
Attn: InsMat

Eastman Kodak Company
50 West Main Street
Rochester 4, New York
2 Attn: Dr. H. Trotter, Jr.
VIA: NIO

General Electric Company
Bldg. #1, Campbell Avenue Plant
Schenectady, New York
1 Attn: Joseph C. Hoffman
VIA: InsMach.

The Rand Corporation
1500 Fourth Street
Santa Monica, California
1 Attn: Librarian
VIA: InsMat

Douglas Aircraft Company, Inc.
3000 Ocean Park Boulevard
Santa Monica, California
1 Attn: E.F. Burton
VIA: BuAero Rep.

North American Aviation, Inc.
12214 Lakewood Boulevard
Downey, California
2 Attn: Aerophysics Library
VIA: BuAero Rep.

National Advisory Committee for Aero
1724 F Street Northwest
Washington 25, D. C.
5 Attn: E. B. Jackson

Ames Aeronautical Laboratory
Moffett Field, California
2 Attn: Dr. A. C. Charter

Langley Aeronautical Laboratory
Langley Field, Virginia
1 Attn: Theoretical Aerodynamics Div.
1 Attn: Dr. A. Busemann
1 Attn: J. Stack

NACA Lewis Flight Propulsion Lab.
Cleveland Hopkins Airport
Cleveland, 11, Ohio
1 Attn: Dr. John C. Evvard

Harvard University
21 Vanserg Building
Cambridge 38, Massachusetts
1 Attn: Prof. G. Birkhoff

Johns Hopkins University
Charles and 34th Streets
Baltimore 18, Maryland
1 Attn: Dr. F. H. Clauser

No. of Copies	
1	New York University 45 Fourth Avenue New York 3, New York Attn: Dr. R. Courant
1	California Institute of Technology Pasadena 4, California Attn: Dr. Hans W. Liepmann
1	University of Toronto Institute of Aerophysics Toronto 5, Ontario Attn: Dr. G.N. Patterson VIA: BuOrd (Ad8)
1	Hughes Aircraft Company Missile Aerodynamics Department Culver City, California Attn: Dr. Allen E. Puckett
1	Institute for Advanced Study Professor John von Neumann Princeton, New Jersey
1	University of Michigan Willow Run Research Center Ypsilanti, Michigan
1	Digital Computer Laboratory University of Illinois Champaign, Illinois Attn: Professor J. P. Nash VIA: ONR
1	International Business Machines Corporation 1111 Connecticut Avenue, N. W. Washington 6, D. C. Attn: Dr. G. W. Petrie

Aeroballistic Research Department
External Distribution List for Applied Math. (X 3a)

No. of Copies	
	Office of Naval Research Branch Office, Navy 100 Fleet Post Office
8	New York, New York
	Commanding General Aberdeen Proving Ground Aberdeen, Maryland
1	Attn: C.L. Poor
1	Attn: Dr. B.L. Hicks
	National Bureau of Standards Washington 25, D. C.
1	Attn. G.B. Schubauer
	Ames Aeronautical Laboratory Moffett Field, California
1	Attn: W.G. Vincenti
	Brown University Providence 12, Rhode Island
1	Attn: Prof. Wm. Prager VIA: Ass't InsMat
	University of California Observatory 21 Berkeley 4, California
1	Attn: Leland E. Cunningham VIA: InsMat
	Cornell University Ithaca, New York
1	Attn: W.P. Sears VIA: ONR
	Indiana University Bloomington, Indiana
1	Attn: T.Y. Thomas VIA: Ass't InsMat
	Massachusetts Institute of Technology 77 Massachusetts Avenue Cambridge 39, Massachusetts
1	Attn: Prof. Eric Reissner VIA: InsMat
	Stanford University Stanford, California
1	Attn: R.J. Langle VIA: Ass't InsMat
	Case Institute of Technology Cleveland, Ohio
1	Attn: Prof. G. Kuerti VIA: ONR
	Harvard University 109 Pierce Hall Cambridge 38, Massachusetts
1	Attn: Prof. R. von Mises